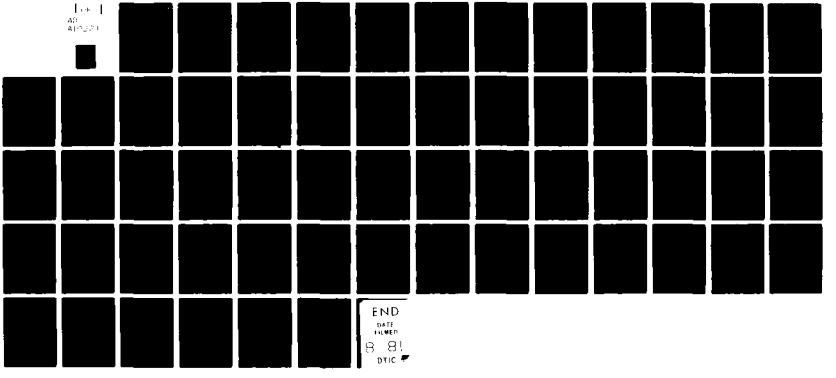
AD-A102 210 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
COMPUTER PROGRAM FOR MANAGEMENT OF A BIBLIOGRAPHIC DATA BASE.(U)
MAY 81 A M BECKA

UNCLASSIFIED DTNSRDC/SME-81-42 NL



END
DATE FILMED
8 81
DTIC

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Maryland 20084

COMPUTER PROGRAM FOR MANAGEMENT OF A BIBLIOGRAPHIC DATA BASE

by

Anne M. Becka

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

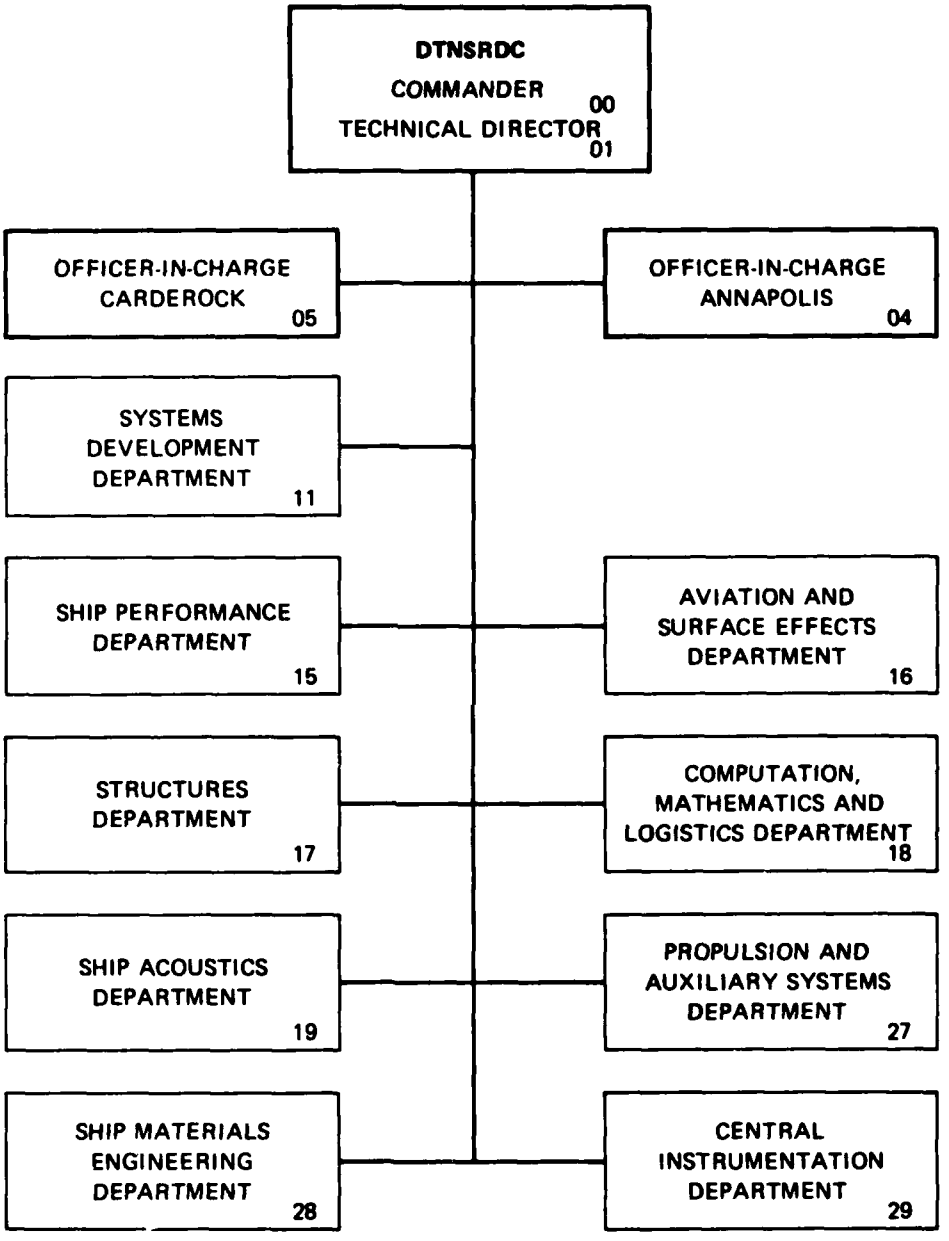
SHIP MATERIALS ENGINEERING DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

May 1981

DTNSRDC/SME-81-42

Computer Program for Management of a Bibliographic Data Base

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DTNSRDC/SME-81-42	2. GOVT ACCESSION NO. AD-A102 210	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer Program for Management of a Bibliographic Data Base		5. TYPE OF REPORT & PERIOD COVERED Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Becka, Anne M.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS David Taylor Naval Ship R&D Center Ship Materials Engineering Department (Code2844) Annapolis, Maryland 21402		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Departmental Overhead
11. CONTROLLING OFFICE NAME AND ADDRESS Ship Materials Engineering Department DTNSRDC/A Annapolis, Maryland 21402		12. REPORT DATE May 1981 13. NUMBER OF PAGES 53
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Indexing, Data Base, Information Retrieval, Batch Processing, Bibliographies

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computerized bibliography allows the indexing and retrieval of scientific research papers, for intense short-term efforts through the use of a large mainframe in the batch mode. Criteria are entered through keypunched cards and the computer produces the requested subsets.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF KEY WORDS

Computer Indexing; Data Base; Information Retrieval; and Batch Processing

ABSTRACT

The computerized bibliography allows the indexing and retrieval of scientific research papers for intense short-term efforts through the use of a large mainframe in the batch mode. Criteria are entered through keypunched cards and the computer produces the requested subsets.

INTRODUCTION

Personnel involved in scientific research often have the need to assemble a bibliography for use in a paper or book or as a tool for advancement of their research. Often this bibliography will cover a broad topic with numerous subtopics identified. This is the situation which precipitated the development of this computer program. Several researchers at this facility were requested to prepare a paper on "Fouling Control Technology". Upon preparation of the outline, 40 initial subtopics were identified. A library computer search supplied the assigned authors with several thousand titles and abstracts applicable to this area. Review of the titles and abstracts revealed a large number of articles worthy of further review and these were obtained. A filing and cross-referencing system was necessary with only about 1 month notice and very limited funds for computer time and personnel.

Due to the limited time and funds, an in-house task was developed to allow organization of the bibliography utilizing the on-site CDC 6600/6700 computing system with a NOS/BE operating system and FORTRAN IV programming language. The program developed may not be the most efficient possibly due to the relatively inexperienced programmer and limited time available, but it proved invaluable for short-term organization of a large number of references. Table 1 is a copy of the outline used for this program. With minor modifications, the program could be adapted to any outline. The program is limited to the number of topics by the configuration of the hardware of the computing system. The articles are cross-referenced by a bit set allowing one bit in a word of memory to represent each section. The CDC 6600/6700 system utilizes a 60 bit word

TABLE 1 - "FOULING CONTROL TECHNOLOGY" OUTLINE

1.0 Introduction

- 1.1 Designing Integrated Fouling Control System

2.0 Evaluation of Efficacy

- 2.1 In Situ Testing
  - 2.1.1 Raft Tests
- 2.2 Accelerated Tests
  - 2.2.1 Dynamic
  - 2.2.2 Bioassay
  - 2.2.3 Leaching Rate

3.0 Chemical Control Technology

- 3.1 Toxic Control Agents
  - 3.1.1 Delivery Systems
    - 3.1.1.1 Coatings
    - 3.1.1.2 Elastomers
    - 3.1.1.3 Direct Injection
    - 3.1.1.4 Impregnation (Wood)
    - 3.1.1.5 Structural Incorporation
- 3.2 Non-Toxic Control Agents
  - 3.2.1 Delivery Systems

4.0 Physical Control Technology

- 4.1 Mechanical Methods of Control
  - 4.1.1 Scrubbing
    - 4.1.1.1 Exterior
    - 4.1.1.2 Interior
  - 4.1.2 Jets
  - 4.1.3 Sonics
    - 4.1.3.1 Ultrasonics
    - 4.1.3.2 Infrasonics
  - 4.1.4 Low Surface Energy Materials
- 4.2 Electrical Methods
- 4.3 Magnetic Methods
- 4.4 Optical Methods
- 4.5 Nuclear Methods
- 4.6 Thermal Methods

TABLE 1 - "FOULING CONTROL TECHNOLOGY" OUTLINE

(continued)

4.7 Osmotic Methods

4.8 Surface Modification Methods

4.9 Explosive Removal Methods

5.0 Conclusions

5.1 Present Practice

5.2 Future Directions

---

and therefore the program can be utilized on this system for up to 60 topics without increasing memory requirements.

PROGRAM CAPABILITIES

This program was designed with limited resources and time. It is for a specific purpose, but many of its methods can be adapted for a wide range of purposes.

The program will read an existing file of references into its working array and then add to it a set of new references. The working array is sorted alphabetically then copied onto a new permanent file. References to all sections requested alphabetically are printed out, then the array is sorted chronologically. References to all sections requested chronologically are printed out, the array is sorted numerically and all sections requested in that order are printed out.

The program requires about 185 CP seconds to run with 550 articles. An array size of 850 references requires 42,000 bytes of memory. For 350 articles about 115 CP seconds are required to run the program and an array size of 1000 references requires 47,000 bytes of memory. A permanent reference file of 550 references takes up 120 PRU's of storage space on the CDC 6600/6700 system.

DATA HANDLING

In the text of this report, a single piece of data is considered to be a single bibliographic reference. Upon receipt of a reference, the authors were instructed to assign the reference a unique accession (access) number. This number was assigned by placing one of preprinted, sequentially numbered labels supplied by the data manager onto the reference.

Upon receipt of 30 references, the authors were prepared to complete a set of data submission sheets. These sheets were specially formatted, standard keypunch forms which correspond to the data entry format in the computer program and are seen as Figures 1, 2, and 3. The author would complete the same line on all three forms for each reference.

Several key instructions were given in completing the forms.

A. Figure 1 (Card 1)

1. Place the accession number as far to the right as possible in the six columns (i.e., 32 becomes ØØØØ32, not 32ØØØØ).
2. Ten spaces are allowed for authors' last names. If a name is longer than 10 letters use nine letters and an asterisk (*) in the tenth space.
3. Put both initials, if known, no periods.
4. In the event a reference has more than 3 authors, put an asterisk (*) in the first space of the area for 3rd author's last name. Leave the rest of the area for 3rd author blank.
5. If an article is by an anonymous author put ANON in the area designated for lst author's last name.
6. In the event that a title is longer than the 36 spaces allowed, put 35 characters and an asterisk (*) in the 36th space.

B. Figure 2 (Card 2)

1. Free area, 38 spaces are allowed for primarily publication information, use ASTI (Applied Science and Technology Index) standard abbreviations for titles, wherever possible.
2. The last 2 digits of the year of publication must be in the columns designated. In the event the year is unknown, put ØØ.

CARD 1

LINE	ACCESSION NUMBER	AUTHOR 1 - LAST NAME	INIT	AUTHOR 2 - LAST NAME	INIT	AUTHOR 3 - LAST NAME	INIT	TITLE
1	6.31	LORENZ	J					ANTIFOULING MEASURES ON SHIPS-A GENER*
2	632	OLDFIELD	D	SANSOM	GF			POTENTIAL ANTIFOULING COATINGS FOR TI*
3	633	BOCKSTEIN*	G	GLEW	G	PHILLIP	AT	UNDERWATER MARINE COATINGS-ELIMINATIO*
4	634	PHILLIP	AT					UNDERWATER MARINE COATINGS- PART I- M*
5	635	DUNN	P	SANSOM	GF			COATED TIMBER FOR UNDERWATER APPLICAT*
6	636	OCHILTREE	BC					ANTIFOULING ELASTOMERIC COMPOSITIONS
7	637	BEITER	CB	HAFNER	LA			AQUEOUS ANTIFOULING COATING COMPOSITI*
8	638	DE FORST	A	PETTIS	RW	PHILLIP	AT	UNDERWATER MARINE COATINGS-A DETAILED*
9	639	RIDENOUR	GM	INGOLS	RS	ARMBRUSTER	EM	SPORICIDAL PROPERTIES OF CHLORINE DIO*
10	640	PORTER	G					END OF THE FREE RIDE
11	641	DEXTER	SC	SULLIVAN	JD*			INFLUENCE OF SUBSTRATE WETTABILITY ON*
12	642	CRISP	DJ					STUDIES OF BARNACLE HATCHING SUBSTANCE
13	643	GERENCSER	VF	BARNOTHY	MF	BARNOTHY	JM	INHIBITION OF BACTERIAL GROWTH BY MAG*
14	644	MITCHELL	R					BIOLOGICAL REPELLANTS: A NEW APPROACH *
15	645	CHADWICK	WL	CLARK	FS	FOX	DL	THERNAL CONTROL OF MARINE FOULING AT *
16	646	WHITE	HE					CONTROL OF MARINE FOULING IN SEAWATER*
17	647	TULLIS	DH	NEILL	LC	HENDERSON	AT	CONTROL OF MARINE ORGANISMS IN A SALT*
18	648	VEAGER	WL	CASTELLI	VJ			ANTIFOULING APPLICATIONS OF VARIOUS T*
19	649	ANDERSEN	DN					ORGANOTIN PRESERVATIVES FOR WOOD STRU*
20	650	EKAHA	HC	VAN LONDEN	AM	DE WOLF	P	RESULTS OF AN INQUIRY INTO THE CONDIT*
21	651	ARNOLD	MH	CLARKE	HJ			PROJECT B-ULTRASONIC ANTIFOULING SHIP*
22	652	ANON						PROGRESS REPORT ON THE TECHNITIUM PRO*
23	653	CHET	I	MITCHELL	R			THE RELATIONSHIP BETWEEN CHEMICAL STR*
24	654	CLARKE	GL					POISONING AND RECOVERY IN BARNACLES A*
25	655	CORNER	EJ	SPARROW				THE MODES OF ACTION OF TOXIC AGENTS-I*
26	656	ANON						THE 'TOXION' SYSTEM-A NEW ANTIFOULING *
27	657	ANON						REDUCING THE BARNACLE BILL
28	658	ANON						SHIPS' HULL PROTECTED-ULTRASONIC VIBR*
29	659	FISK	NR					A VIEW OF ANTIFOULING
30	660	FLEMING	H					EFFECT OF HIGH FREQUENCY FIELDS ON MI*

FIGURE 1 - CARD 1

CARD 2

FREE FORMAT SOURCE PRINTER DATA

		YEAR
1	AD-B032 858	78
2	AD-918 043	73
3	AD-B006 675	74
4	AD-907 612	72
5	AD-902 136	72
6	AD-911 382	72
7	US PATENT # 4,052,354	77
8	AD-922 986	74
9	WATER SEWAGE WORKS, V96, N8, P279-83, A*	49
10	NBS DIMENSIONS, V64, N2, P12-7, 1980	80
11	APP MICROBIOL, V30, N2, P298-308, AUG 19	75
12	COMP BIOCHEM PHYSIOL, V30, N6, P1037-4*	69
13	NATURE, V196, N485, P539-41, NOV 10, 1962	62
14	NAVSEA JOUR, 62-6, JULY 1976	76
15	TRANS ASME, V72, P127-31, FEB 1950	50
16	TRANS ASME, V22, P117-26, FEB 1950	50
17	J INST PET, V46, N426, P155-67, JUN 1960	59
18	ORGANOMETALLIC POLYMERS, ED CE CARRA*	77
19	DTNSRDC REPORT #SME-78/41, JUNE 1979	79
20	TNO REPORT 47C,	62
21	TECHNICAL MINUTE #93	52
22	UNIV OF VIRGINIA ALUM PATENTS FOUND*	79
23	CAN. J. MICROBIOL, V22, P1206-08, N8, 1976	76
24	BIOLOGICAL BULL, V92, P73-91, 1947	47
25	J MAR BIOL ASS UK, V35, P531-48, 1956	56
26	CORR PREV & CONT, P49-54, MARCH 1960	60
27	CHEM WEEK, V72, N9, 87-91, FEB 28, 1963	53
28	ENGINEERING, V180, P416, SEPT 23, 1955	55
29	PAINT TECH, V24, N270, P15-18, MAY 1960	60
30	ELEC ENG, P18-21, JAN 1949	44

FIGURE 2 - CARD 2

CARD 3 (PLACE A 1 UNDER THE APPROPRIATE COLUMN(S))

1000 INTRODUCTION
1100 DESIGNING INTEGRATED FOULING CONTROL SYSTEM
2000 EVAL. OF EFFICACY
2100 IN SITU TESTING
2110 RAFT TESTS
2200 ACCELERATED TESTS
2210 DYNAMIC
2220 BIOASSAY
2230 LEACHING RATE
3000 CHEMICAL CONTROL TECHNOLOGY
3100 TOXIC CONTROL AGENTS
3110 DELIVERY SYSTEMS
3111 COATINGS
3112 ELASTOMERS
3113 DIRECT INJECTION
3114 IMPREGNATION (WOOD)
3115 STRUCTURAL INCORPORATION (GRP CONCRETE)
3200 NON-TOXIC CONTROL AGENTS
3210 DELIVERY SYSTEMS
4000 PHYSICAL CONTROL TECHNOLOGY
4100 MECHANICAL METHODS OF CONTROL
4110 SCRUBBING
4111 EXTERIOR (FISHNET)
4112 INTERIOR (FLAGELLATION LINE)
4120 JETS
4130 SONICS
4131 ULTRASONICS
4132 INFRASONICS
4140 LOW SURFACE ENERGY MATERIALS
4200 ELECTRICAL METHODS
4300 MAGNETIC METHODS
4400 OPTICAL METHODS
4500 NUCLEAR METHODS
4600 THERMAL METHODS (STEAM PURGE)
4700 OSMOTIC METHODS
4800 SURFACE MOD. METHODS
4900 EXPLOSIVE REMOVAL METHODS
5000 CONCLUSIONS
5100 PRESENT PRACTICE
5200 FUTURE DIRECTIONS

FIGURE 3 - CARD 3

C. Figure 3 (Card 3)

1. Place a 1 (one) in the columns corresponding to the sections under which the reference is useful.
2. Each article should be designated under as broad a range of categories as possible (i.e., if an article is applicable to section 4112, it would be anticipated to also be applicable to sections 4110, 4100, and 4000).

D. General Instructions

1. Use all capital letters in filling out the data sheets.

Upon completion of a set of data sheets, the articles were filed and the sheets were sent for keypunching.

Data sheet design was based on several constraints of the program. We were hoping to limit memory usage and run time, and provide access to data lines by NETED, (text editor modeled after the standard Arpanet Editor). The data sheets proved rather tedious to complete, but suited the purpose of the program ideally.

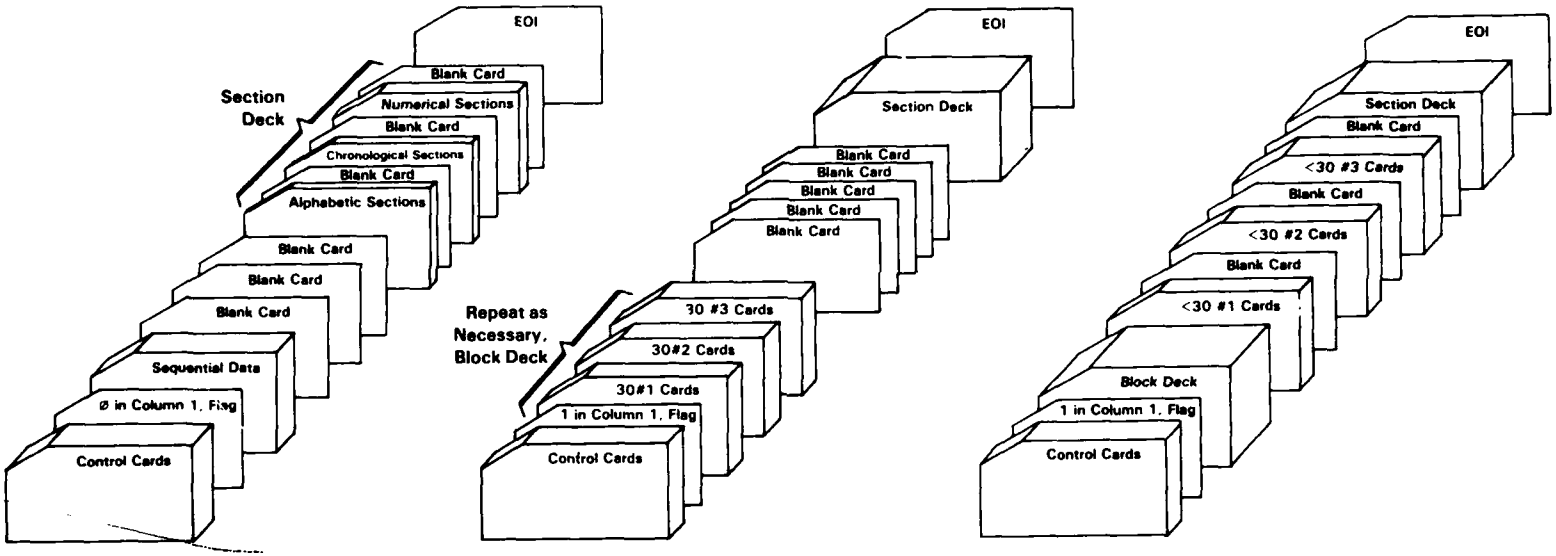
DATA INPUT

Data (references) can be input in two ways. One is to submit groups of up to 30 first cards, followed by the same number of second, then third cards. This is referred to as block input.

The second method is to submit the cards sequentially i.e., the three cards for each reference are together. The deck of data cards is preceded by a single card with a flag which indicates the order the cards are in. A flag of zero (Ø) in the 1st column represents sequential input and a flag of one (1) represents block input. Blank cards within the deck represents the end of the input of data.

Upon reading in the new data the program will output any sections requested with the references in alphabetical order, then any sections requested with the references in chronological order, and finally any sections requested with the references in numerical order.

Various possible input decks are illustrated in Table 2. An example block input and the resulting output is in Appendix B.



Sequential Data Input Deck

Block Data Input With 30 References in Each Block

Block Data Input With Less Than 30 References in Last Block

TABLE 2

Sections are referenced by four digit section numbers corresponding to the subtopic numbers seen in Table 1, (i.e., 4.1 becomes 4100, 3.1.1.2 becomes 3112).

PROGRAM LOGIC

The program logic can best be illustrated through the use of Figure 4. The various abbreviations are defined below:

A. BIBLIO - The title of the source program.

B. PREFS - The permanent file containing the old references.

C. INPUT - The card deck containing new references and sections desired.

D. RDFLE - The subroutine which reads PREFS into the working array.

E. READIN - The subroutine which reads in sequential data.

F. RDBLK - The subroutine which reads in block data.

G. BINDEX - The subroutine which computes the index corresponding to the sections identified on the data sheets.

H. SORT - The subroutine which sorts the working array into alphabetical order by 1st author's last name.

I. DUPS - Subroutine which checks 1st author's last name and year for duplicates.

J. WRTOUT - The subroutine which writes out a reference onto output.

K. WRTFLE - The subroutine which writes out a copy of the working array onto NEWREF.

L. NEWREF - An alphabetical file containing an updated version of the full reference list (to be changed to PREFS, by operator, after program is complete).

M. PRNT - The subroutine which prints out all the references identified for any one section.

N. BINBAC - The subroutine which back calculates the sections from the index.

O. SORTYR - The subroutine which arranges the working array in chronological order.

P. SORTAC - The subroutine which arranges the working array in numerical order by access number.

FIGURE 4

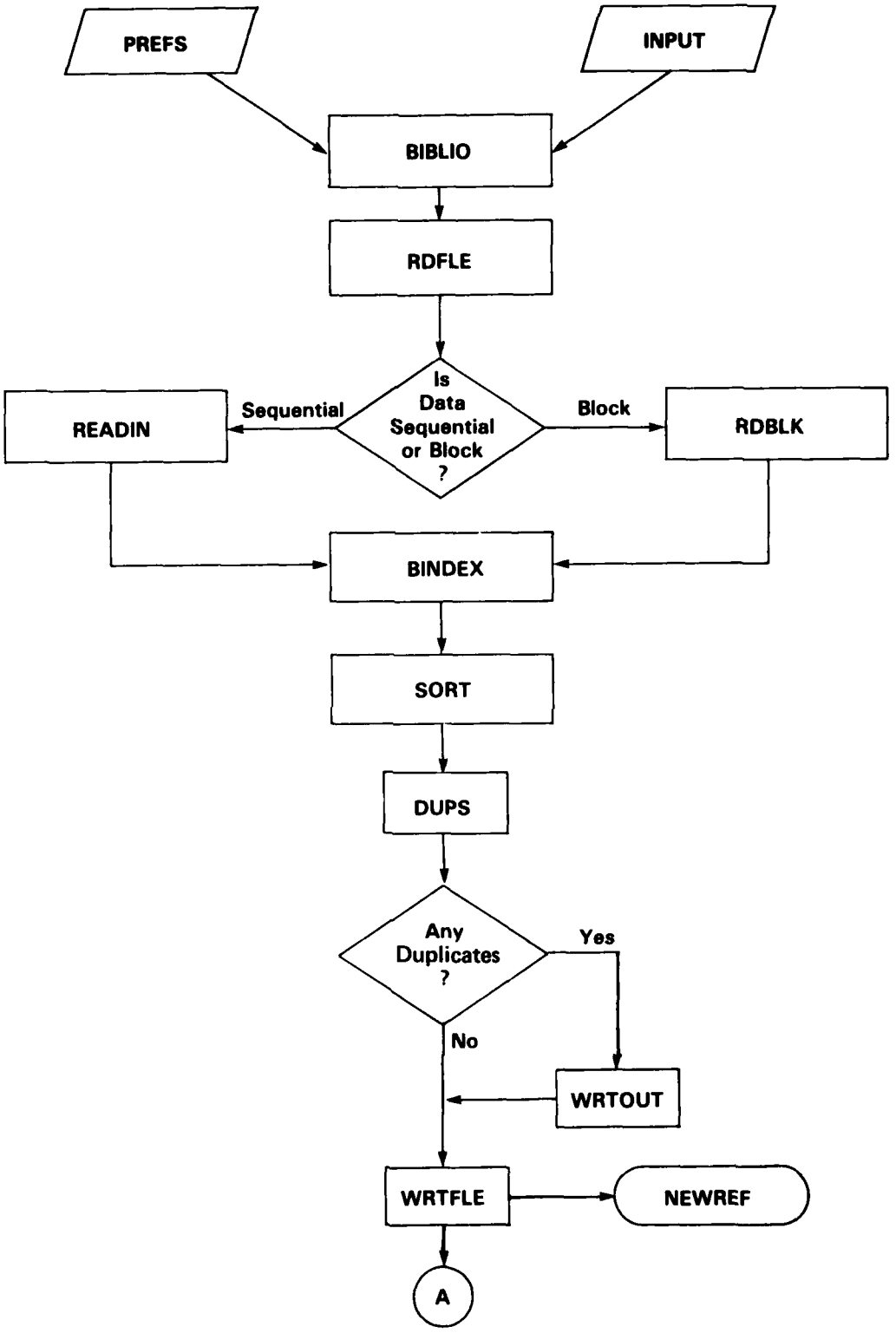
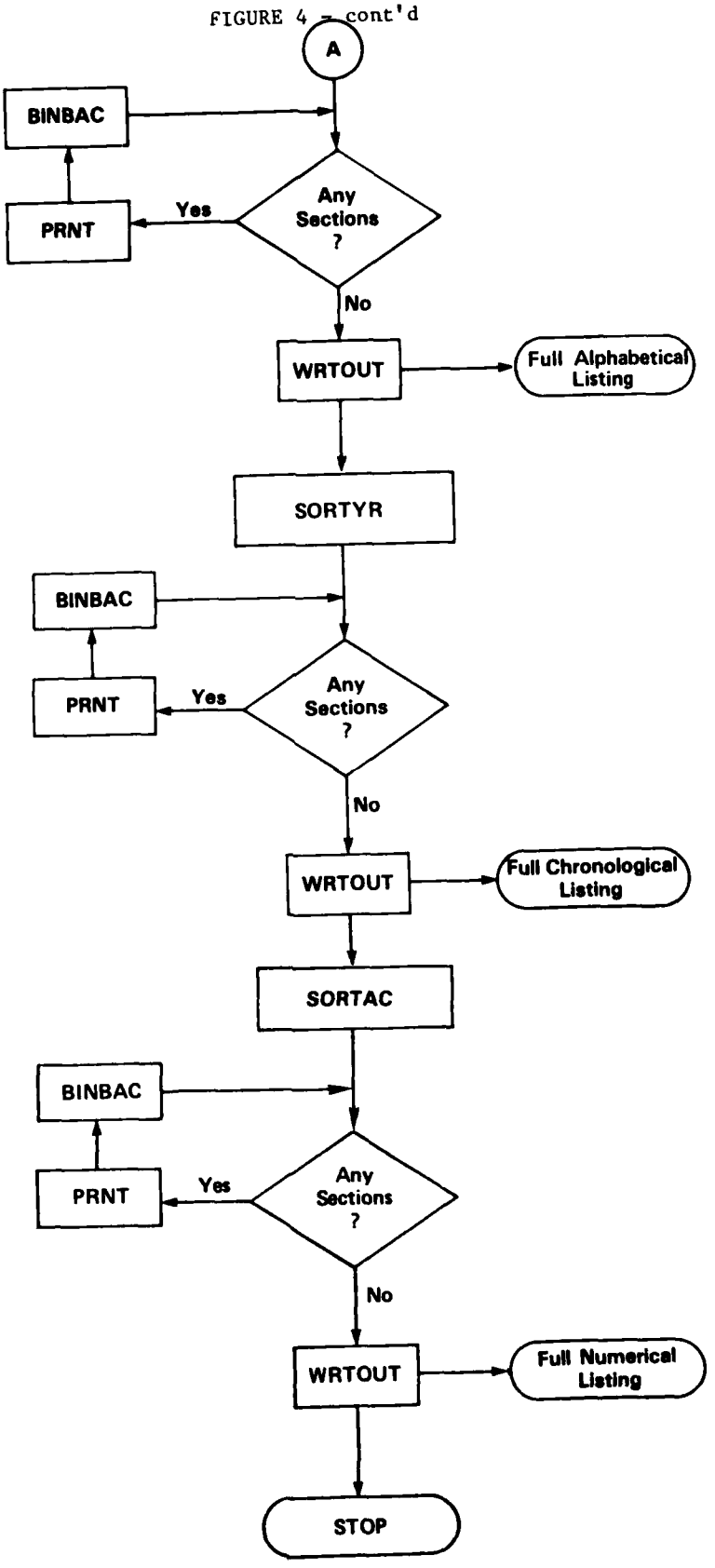

PREFS
INPUT
BIBLIO
RDFLE
Is Data Sequential or Block ?
Sequential
Block
READIN
RDBLK
BINDEX
SORT
DUPS
Any Duplicates ?
Yes
No
WRTOUT
WRTFLE
NEWREF
A

FIGURE 4 - cont'd



COMPUTER SYSTEM ENVIRONMENT

HARDWARE

The hardware configuration of DTNSRDC's CDC 6600/6700 is as follows:

- o Dual central processors, 131,072 60-bit word memory each,
- o 20 peripheral processors for each central processor,
- o Model 844 disk drives,
- o One 1700 terminal, and
- o Two 2550 data concentrators

SOFTWARE

The software operating system installed for the computer is the CDC NOS/BE v. 1.4 operating system. A typical control deck for this program is as follows:

- o REQUEST,NEWREF,*PF.
- o ATTACH,PREFS,ID=XXXX.
- o ATTACH,BIBLIO,ID=XXXX.
- o FIN,I=BIBLIO,SL=0,R=0,PD=8.
- o LGO,PL=1ØØØØØ.
- o CATALOG,NEWREF,PREFS,ID=XXXX.
- o PURGE,PREFS,PREFS,ID=XXXX.

(XXXX specifies user's registered computer initials.)

This format allows for the Fortran source code to be maintained on the computer (BIBLIO) as well as the permanent reference data base (PREFS). Old versions of PREFS are purged as soon as the file is updated. The sorting of the data base is done by the source program and the data base is edited interactively using the available on-line editor NETED v. 1.4.

APPENDIX A

SOURCE LISTING

```
      PROGRAM BIBLIO (INPUT,OUTPUT,PREFS,NEWREF,TAPE5=INPUT,TAPE6
     1=OUTPUT,TAPE8=NEWREF,TAPE9=PREFS)
C
C*****************************************************************
C
C     THIS PROGRAM IS DESIGNED TO TAKE A LARGE AMOUNT OF
C     BIBLIOGRAPHIC DATA, SORT IT,CHECK FOR DUPLICATES AND BE ABLE TO
C     OUTPUT CERTAIN CATEGORIES OF REFERENCES.
C
C*****************************************************************
C
      COMMON LIST (1000,17),INDEX (40),ITWOS(40)
      DO 2 I=1,40
      N=I-1
      ITWOS(I)=2**N
    2 CONTINUE
      J=0
C
C*****************************************************************
C
C  READ IN PERMANENT REFERENCE FILE (PREFS) INTO PROGRAM
C  OPERATING ARRAY.
C
C*****************************************************************
C
      CALL RDFLE (J)
      READ (5,1) IFLAG
    1 FORMAT (I1)
C
C*****************************************************************
C
C  EVALUATE FLAG TO SEE WHAT FORMAT NEW BIBLIOGRAPHIC ENTRIES
C  ARE IN.  1 INDICATES BLOCK FORMAT, 0 INDICATES SEQUENTIAL
C  FORMAT.
C
C*****************************************************************
C
      IF (IFLAG.EQ.0) CALL READIN (J)
      IF (IFLAG.EQ.1) CALL RDBLK (J)
      DO 80 L6=1,3
C
C*****************************************************************
C
C  PERFORM ALPHABETICAL SORT IF THIS IS THE FIRST TIME IN THE LOOP
C
C*****************************************************************
C
      IF (L6.EQ.1)CALL SORT (J)
C
C*****************************************************************
C
C  PERFORM CHRONOLOGICAL SORT IF THIS IS THE SECOND TIME IN THE
C  LOOP.
C
C*****************************************************************
C
      IF (L6.EQ.2) CALL SORTYR (J)
```

```
C
C*************************************************************************
C
C  PERFORM NUMERICAL SORT BY ACCESS NUMBER IF THIS IS THE THIRD
C  TIME IN THE LOOP.
C
C*************************************************************************
C
      IF (L6.EQ.3) CALL SORTAC (J)
      CONTINUE
      DO 30 L=1,50
      READ (5,120) ISECT
C
C*************************************************************************
C
C  READ IN THE SECTION TO BE WRITTEN OUT.
C  IF SECTION IS EQUAL TO 00 GO TO THE TOP OF THE LOOP AND PERFORM
C  THE NEXT SORT OR FINISH OUT THE PROGRAM IF THE LAST SORT
C  PERFORMED WAS THE NUMERICAL SORT.
C
C*************************************************************************
C
  120 FORMAT (I4)
      IF (ISECT.EQ.0) GO TO 50
      CALL PRNT (ISECT,J)
   30 CONTINUE
   50 CONTINUE
      WRITE (6,100)
  100 FORMAT ('1','FULL REFERENCE LIST',//)
      DO 40 K=1,J
C
C*************************************************************************
C
C  WRITE OUT THE FULL REFERENCE LIST TO GIVE A LISTING OF ALL
C  REFERENCE AVAILABLE IN THE ORDER OF THE LATEST SORT.
C
C*************************************************************************
C
      CALL WRTOUT (K)
   40 CONTINUE
   80 CONTINUE
      STOP
      END
```

```
      SUBROUTINE BINBAC (I)
C
C*********************************************************************
C
C     THIS SUBROUTINE BRINGS BACK THE SECTION CODE FROM THE INDEX
C     CALCULATED IN SUBROUTINE BINDEX
C
C*********************************************************************
C
      COMMON LIST (1000,17), INDEX(40),ITWOS(40)
      P=LIST (I,17)
      DO 5 I1=1,40
      INDEX (I1)=0
    5 CONTINUE
      DO 15 K=1,40
      L=41-K
      M=P-ITWOS(L)
      IF (M.LT.0) GO TO 25
      INDEX(L)=1
      P=M
   25 CONTINUE
   15 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BINDEX (M)
C
C*****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE INDEX WHICH DETERMINES TO WHICH
C     OF THE 40 CATEGORIES THE ENTRY APPLIES
C
C*****************************************************************
C
      COMMON LIST (1000,17), INDEX (40),ITWOS(40)
      IND=0
      DO 5 K=1,40
      IF (INDEX(K).EQ.1) IND=IND+ITWOS(K)
    5 CONTINUE
      LIST (M,17)=IND
      RETURN
      END
```

```
      SUBROUTINE DUPS (N)
C
C*********************************************************************
C
C     THIS SUBROUTINE CHECKS THE REFERENCE ARRAY FOR DUPLICATES
C
C*********************************************************************
C
      COMMON LIST (1000,17),INDEX(40),ITWOS(10)
      WRITE (6,1)
    1 FORMAT ('1','POTENTIAL DUPLICATES INCLUDE:',//)
      I=0
    5 CONTINUE
      I=I+1
      DO 10 I1=I,N
      I2=I1+1
      IF (LIST(I,2).NE.LIST(I2,2)) GO TO 20
   10 CONTINUE
   20 CONTINUE
      I3=I2-2
      IF (I.GE.N) RETURN
      IF (I.GT.I3) GO TO 15
      DO 30 I4=I,I3
      I5=I4+1
      IF (LIST (I,16).NE.LIST(I5,16)) GO TO 25
      CALL WRTOUT (I)
      CALL WRTOUT (I5)
   25 CONTINUE
   30 CONTINUE
      I=I+1
      IF (I.GE.I3) GO TO 15
      GO TO 20
   15 I=I3
      I=I+1
      GO TO 5
      END
```

```
      SUBROUTINE PRNT (IN,N)
C
C*****************************************************************************
C
C     THIS SUBROUTINE WILL DETERMINE WHICH OF THE SECTIONS ARE
C     DESIRED AND CONVERT THE SECTION NUMBER INTO A FORM UNDERSTOOD
C     BY THE INDEX CALCULATION TO DETERMINE THE NEEDED REFERENCES
C
C*****************************************************************************
C
      COMMON LIST (1000,17),INDEX(40),ITWOS(40)
      IF (IN.EQ.1000) GO TO 10
      IF (IN.EQ.1100) GO TO 20
      IF (IN.EQ.2000) GO TO 30
      IF (IN.EQ.2100) GO TO 40
      IF (IN.EQ.2110) GO TO 50
      IF (IN.EQ.2200) GO TO 60
      IF (IN.EQ.2210) GO TO 70
      IF (IN.EQ.2220) GO TO 80
      IF (IN.EQ.2230) GO TO 90
      IF (IN.EQ.3000) GO TO 100
      IF (IN.EQ.3100) GO TO 110
      IF (IN.EQ.3110) GO TO 120
      IF (IN.EQ.3111) GO TO 130
      IF (IN.EQ.3112) GO TO 140
      IF (IN.EQ.3113) GO TO 150
      IF (IN.EQ.3114) GO TO 160
      IF (IN.EQ.3115) GO TO 170
      IF (IN.EQ.3200) GO TO 180
      IF (IN.EQ.3210) GO TO 190
      IF (IN.EQ.4000) GO TO 200
      IF (IN.EQ.4100) GO TO 210
      IF (IN.EQ.4110) GO TO 220
      IF (IN.EQ.4111) GO TO 230
      IF (IN.EQ.4112) GO TO 240
      IF (IN.EQ.4120) GO TO 250
      IF (IN.EQ.4130) GO TO 260
      IF (IN.EQ.4131) GO TO 270
      IF (IN.EQ.4132) GO TO 280
      IF (IN.EQ.4140) GO TO 290
      IF (IN.EQ.4200) GO TO 300
      IF (IN.EQ.4300) GO TO 310
      IF (IN.EQ.4400) GO TO 320
      IF (IN.EQ.4500) GO TO 330
      IF (IN.EQ.4600) GO TO 340
      IF (IN.EQ.4700) GO TO 350
      IF (IN.EQ.4800) GO TO 360
      IF (IN.EQ.4900) GO TO 370
      IF (IN.EQ.5000) GO TO 380
      IF (IN.EQ.5100) GO TO 390
      IF (IN.EQ.5200) GO TO 400
      WRITE (6,1) IN
    1 FORMAT ('1','SECTION ',I4,' IS NON-EXISTENT, CHECK OUTLINE')
      RETURN
```

```
 10 WRITE (6,2)
  2 FORMAT ('1','INTRODUCTION (DEFINITION OF MARINE FOULING) SECTION
   1 REFERENCES',//)
    I=1
    GO TO 6000
 20 WRITE (6,3)
  3 FORMAT ('1','DESIGN INTEGRATED FOULING CNTRL SYSTS REFERENCES',//)
    I=2
    GO TO 6000
 30 WRITE (6,4)
  4 FORMAT ('1','EVAL OF EFFICACY SECTION REFERENCES',//)
    I=3
    GO TO 6000
 40 WRITE (6,5)
  5 FORMAT ('1','IN SITU TESTING SECTION REFERENCES',//)
    I=4
    GO TO 6000
 50 WRITE (6,6)
  6 FORMAT ('1','RAFT TESTS SECTION REFERENCES',//)
    I=5
    GO TO 6000
 60 WRITE (6,7)
  7 FORMAT ('1','ACCELERATED TESTS SECTION REFERENCES',//)
    I=6
    GO TO 6000
 70 WRITE (6,8)
  8 FORMAT ('1','DYNAMIC SECTION REFERENCES',//)
    I=7
    GO TO 6000
 80 WRITE (6,9)
  9 FORMAT ('1','BIOASSAY SECTION REFERENCES',//)
    I=8
    GO TO 6000
 90 WRITE (6,11)
 11 FORMAT ('1','LEACHING RATE SECTION REFERENCES',//)
    I=9
    GO TO 6000
100 WRITE (6,12)
 12 FORMAT ('1','CHEM CONTROL TECHNOLOGY SECTION REFERENCES',//)
    I=10
    GO TO 6000
110 WRITE (6,13)
 13 FORMAT ('1','TOXIC CNTRL AGENTS SECTION REFERENCES',//)
    I=11
    GO TO 6000
120 WRITE (6,14)
 14 FORMAT ('1','DELIVERY SYSTEMS SECTION REFERENCES',//)
    I=12
    GO TO 6000
130 WRITE (6,15)
 15 FORMAT ('1','COATINGS SECTION REFERENCES',//)
    I=13
    GO TO 6000
```

```
140 WRITE (6,16)
 16 FORMAT('1','ELASTOMERS SECTION REFERENCES',//)
    I=14
    GO TO 6000
150 WRITE (6,17)
 17 FORMAT ('1','DIRECT INJECTION SECTION REFERENCES',//)
    I=15
    GO TO 6000
160 WRITE (6,18)
 18 FORMAT ('1','IMPREGNATION (WOOD) SECTION REFERENCES',//)
    I=16
    GO TO 6000
170 WRITE (6,19)
 19 FORMAT ('1','STRUCTURAL INCORPORATION SECTION REFERENCES',//)
    I=17
    GO TO 6000
180 WRITE (6,21)
 21 FORMAT ('1','NON-TOXIC CONTROL AGENTS SECTION REFERENCES',//)
    I=18
    GO TO 6000
190 WRITE (6,22)
 22 FORMAT ('1','DELIVERY SYSTEMS SECTION REFERENCES',//)
    I=19
    GO TO 6000
200 WRITE (6,23)
 23 FORMAT ('1','PHYSICAL CONTROL TECHNOLOGY SECTION REFS',//)
    I=20
    GO TO 6000
210 WRITE (6,24)
 24 FORMAT ('1','MECH METHODS OF CNTRL SECTION REFERENCES',//)
    I=21
    GO TO 6000
220 WRITE (6,25)
 25 FORMAT ('1','SCRUBBING SECTION REFERENCES',//)
    I=22
    GO TO 6000
230 WRITE (6,26)
 26 FORMAT ('1','EXTERIOR SECTION REFERENCES',//)
    I=23
    GO TO 6000
240 WRITE (6,27)
 27 FORMAT ('1','INTERIOR SECTION REFERENCES',//)
    I=24
    GO TO 6000
250 WRITE (6,28)
 28 FORMAT ('1','JETS SECTION REFERENCES',//)
    I=25
    GO TO 6000
260 WRITE (6,29)
 29 FORMAT ('1','SONICS SECTION REFERENCES',//)
    I=26
    GO TO 6000
270 WRITE (6,31)
 31 FORMAT ('1','ULTRASONICS SECTION REFERENCES',//)
    I=27
    GO TO 6000
```

```
280 WRITE (6,32)
 32 FORMAT ('1','INFRASONICS SECTION REFERENCES',//)
    I=28
    GO TO 6000
290 WRITE (6,33)
 33 FORMAT ('1','LOW SURFACE ENRGY SECTION REFERENCES',//)
    I=29
    GO TO 6000
300 WRITE (6,34)
 34 FORMAT ('1','ELECTRICAL METHODS SECTION REFERENCES',//)
    I=30
    GO TO 6000
310 WRITE (6,35)
 35 FORMAT ('1','MAGNETIC METHODS SECTION REFERENCES',//)
    I=31
    GO TO 6000
320 WRITE (6,36)
 36 FORMAT ('1','OPTICAL METHODS SECTION REFERENCES',//)
    I=32
    GO TO 6000
330 WRITE (6,37)
 37 FORMAT ('1','NUCLEAR METHODS SECTION REFERENCES',//)
    I=33
    GO TO 6000
340 WRITE (6,38)
 38 FORMAT ('1','THERMAL METHODS SECTION REFERENCES',//)
    I=34
    GO TO 6000
350 WRITE (6,39)
 39 FORMAT ('1','OSMOTIC METHODS SECTION REFERENCES',//)
    I=35
    GO TO 6000
360 WRITE (6,41)
 41 FORMAT ('1','SURFACE MOD METHODS SECTION REFERENCES',//)
    I=36
    GO TO 6000
370 WRITE (6,42)
 42 FORMAT ('1','EXPLOSIVE REMOVAL SECTION REFERENCES',//)
    I=37
    GO TO 6000
380 WRITE (6,43)
 43 FORMAT ('1','CONCLUSIONS SECTIONS REFERENCES',//)
    I=38
    GO TO 6000
390 WRITE (6,44)
 44 FORMAT ('1','PRESENT PRACTICE SECTION REFERENCES',//)
    I=39
    GO TO 6000
400 WRITE (6,45)
 45 FORMAT ('1','FUTURE DIRECTIONS SECTION REFERENCES',//)
    I=40
6000 CONTINUE
```

```
      WRITE (6,6911)
6911  FORMAT (1X,'ACCESS    FIRST',10X,'SECOND',8X,'THIRD',9X,'TITLE',
     136X,'PUBLISHER',25X,'YEAR')
      WRITE (6,6912)
6912  FORMAT (1X,'NUMBER    AUTHOR',9X,'AUTHOR',8X,'AUTHOR',///)
      DO 7000 I1=1,N
      CALL BINBAC (I1)
      IF (INDEX(I).EQ.0) GO TO 6999
      CALL WRTOUT (I1)
6999  CONTINUE
7000  CONTINUE
      WRITE (6,7001)
7001  FORMAT (////)
      RETURN
      END
```

```
      SUBROUTINE RDBLK (J)
C
C*************************************************************************
C
C     THIS SUBROUTINE WILL READ IN THE DATA IN BLOCKS OF 30 CARDS
C
C*************************************************************************
C
      COMMON LIST (1000,17),INDEX(40),ITWOS(40)
      DO 10 I=1,500
      N=0
      DO 20 I1=1,30
      J=J+1
      READ (5,100)(LIST(J,J1),J1=1,11)
  100 FORMAT (I6,A10,A2,A10,A2,A10,A2,3A10,A8)
      IF (LIST (J,1).EQ.0) GO TO 25
      N=N+1
   20 CONTINUE
   25 CONTINUE
      I8=J-N+1
      K3=J
      IF (N.LT.30) I8=I8-1
      IF (N.LT.30) K3=K3+1
      DO 30 I2=I8,K3
      READ (5,110)(LIST(I2,J2),J2=12,16)
  110 FORMAT (3A10,A6,I2)
   30 CONTINUE
      DO 40 I3=I8,K3
      READ (5,120)(INDEX(K5),K5=1,40)
  120 FORMAT (40I2)
      CALL BINDEX (I3)
   40 CONTINUE
      IF (N.LT.30) GO TO 15
   10 CONTINUE
   15 J=J-1
      RETURN
      END
```

```
      SUBROUTINE RDFLE (J)
C
C*****************************************************************************
C
C     THIS SUBROUTINE READS A FILE CALLED REFS INTO THE
C     PROGRAM OPERATING ARRAY
C
C*****************************************************************************
C
      COMMON LIST (1000,17),INDEX (40),ITWOS(40)
      DO 5 I=1,1000
      READ (9,100)(LIST (I,K),K=1,17)
  100 FORMAT (I6,A10,A2,A10,A2,A10,A2,3A10,A8,3A10,A6,I2,I13)
      IF (LIST(I,1).EQ.0) RETURN
      J=J+1
    5 CONTINUE
      END
```

```
      SUBROUTINE READIN (J)
C
C*************************************************************************
C
C     THIS SUBROUTINE READS IN EACH OF THE BIBLIOGRAPHIC ENTRIES
C
C*************************************************************************
C
      COMMON LIST (1000,17),INDEX (40),ITWOS(40)
      K1=J+1
      DO 10 I=K1,1000
      READ (5,100)(LIST(I,J1),J1=1,16)
  100 FORMAT (I6,A10,A2,A10,A2,A10,A2,3A10,A8,/,3A10,A6,I2)
      READ (5,110) (INDEX(K),K=1,40)
  110 FORMAT (40I2)
      IF (LIST(I,1).EQ.0) RETURN
      J=J+1
      CALL BINDEX (I)
   10 CONTINUE
      END
```

```
      SUBROUTINE SORT (N)
C
C*****************************************************************************
C
C     THIS SUBROUTINE SORTS THE ENTRIES INTO ALPHEBETICAL ORDER AND
C     CHECKS FOR DUPLICATES
C
C*****************************************************************************
C
      COMMON LIST (1000,17),INDEX (40),ITWOS(40)
      M1=N
   25 I2=0
      M1=M1-1
      DO 5 I=1,M1
      I1=LIST (I,2)
      J=I+1
      I3=LIST (J,2)
      IF (I3-I1) 10,20,20
   10 DO 15 K=1,17
      M=LIST(I,K)
      LIST (I,K)=LIST(J,K)
      LIST (J,K)=M
   15 CONTINUE
   20 I2=I2+1
    5 CONTINUE
      IF (I2.GT.0) GO TO 25
      CALL DUPS (N)
      CALL WRTFLE (N)
      RETURN
      END
```

```
      SUBROUTINE SORTAC (N)
C
C*************************************************************************
C
C     THIS SUBROUTINE SORTS THE REFERENCES BY
C      ACCESS NUMBER
C
C*************************************************************************
C
      COMMON LIST (1000,17),INDEX(40),ITWOS(40)
      M1=N
   25 I2=0
      M1=M1-1
      DO 5 I=1,M1
      I1=LIST(I,1)
      J=I+1
      I3=LIST(J,1)
      IF (I3-I1) 10,20,20
   10 DO 15 K=1,17
      M=LIST (I,K)
      LIST (I,K)=LIST(J,K)
      LIST (J,K)=M
   15 CONTINUE
      I2=I2+1
   20 CONTINUE
    5 CONTINUE
      IF (I2.GT.0) GO TO 25
      RETURN
      END
```

```
      SUBROUTINE SORTYR (N)
C
C*****************************************************************************
C
C     THIS SUBROUTINE SORTS THE REFERENCES BY YEAR
C
C*****************************************************************************
C
      COMMON LIST (1000,17),INDEX (40),ITWOS(40)
      M1=N
   25 I2=0
      M1=M1-1
      DO 5 I=1,M1
      I1=LIST (I,16)
      J=I+1
      I3=LIST (J,16)
      IF (I3-I1) 10,20,20
   10 DO 15 K=1,17
      M=LIST (I,K)
      LIST (I,K)=LIST (J,K)
      LIST (J,K)=M
   15 CONTINUE
      I2=I2+1
   20 CONTINUE
    5 CONTINUE
      IF (I2.GT.0) GO TO 25
      RETURN
      END
```

```
      SUBROUTINE WRTOUT (I1)
C
C*********************************************************************
C
C     THIS SUBROUTINE WRITES OUT ANY REFERENCE ON THE OUTPUT
C     DEVICE.
C
C*********************************************************************
C
      COMMON LIST (1000,17), INDEX (40),ITWOS(40)
      WRITE (6,6998) (LIST(I1,L),L=1,16)
 6998 FORMAT ('0',I5,1X,A10,',',A2,';',1X,A10,',',A2,';',1X,A10,
     1',',A2,';',1X,3A10,A8,3X,3A10,A6,2X,I2)
      RETURN
      END
```

```
      SUBROUTINE WRTFLE (N)
C
C*********************************************************************
C
C     THIS SUBROUTINE WRITES OUT THE REFERENCES IN ALPHABETICAL
C     ORDER ONTO A PERMANENT FILE
C
C*********************************************************************
C
      COMMON LIST (1000,17),INDEX(40),ITWOS(40)
      N1=N+1
      DO 5 I=1,N1
      WRITE (8,100) (LIST (I,J),J=1,17)
  100 FORMAT (I6,A10,A2,A10,A2,A10,A2,3A10,A8,3A10,A6,I2,I13)
    5 CONTINUE
      RETURN
      END
```

APPENDIX B

SAMPLE INPUT AND OUTPUT

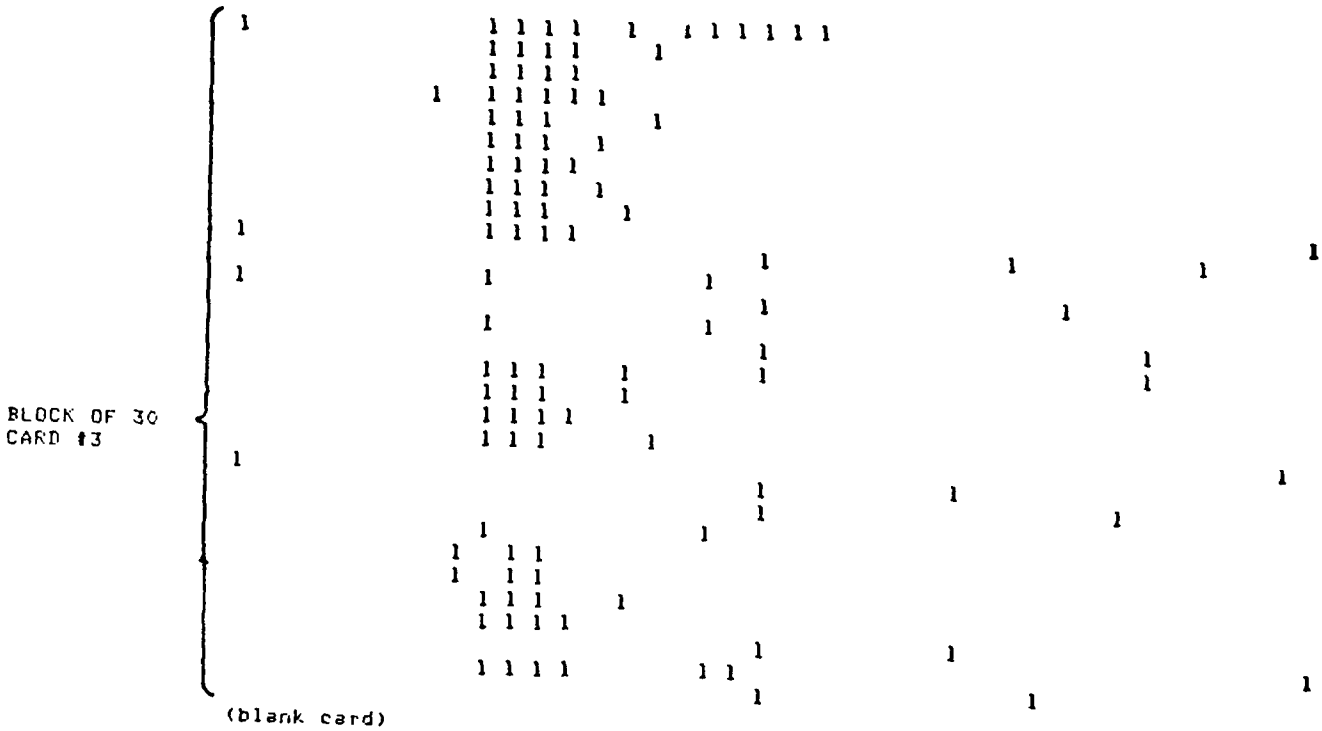
1

BLOCK OF 30
CARD #1

631LORENZ J ANTIFOULING MEASURES ON SHIPS-A GENER*
6320LDFIELD D SANSOM GF POTENTIAL ANTIFOULING COATINGS FOR II*
633BOCKSTEIN*G GLEW G PHILLIP ATUNDERWATER MARINE CATINGS-ELIMINATION*
634PHILLIP AT UNDERWATER MARINE COATINGS- PART I- M*
6350UNN P SANSOM GF COATED TIMBER FOR UNDERWATER APPLICAT*
6360CHILTREE BC ANTIFOULING ELASTOMERIC COMPOSITIONS
637BEITEP CBHAFNER LA AQUEOUS ANTIFOULING COATING COMPOSITI*
638DE FORST A PETTIS RWPHILLIP ATUNDERWATER MARINE COATINGS-A DETAILED*
639RIDENOUR GMINGOLS RSARMBRUSTEREMSPORICIDAL PROPERTIES OF CHLORINE DIO*
640PORTER G END OF THE FREE RIDE
641DEXTER SCSULLIVAN JD* INFLUENCE OF SUBSTRATE WETTABILITY ON*
642CRISP DJ STUDIES OF BARNACLE HATCHING SUBSTANCE
643GERENCSER VFBARNOTHY MFBARNOTHY JMINHIBITION OF CACTERIAL GROWTH BY MAG*
644MITCHELL R BIOLOGICAL REPELLENTS:A NEW APPROACH *
645CHADWICK WLCLARK FSFOX DLTHERMAL CONTROL OF MARINE FOULING AT *
646WHITE HE CONTROL OF MARINE FOULING IN SEA-WATE*
647TULLIS DHNEILL LCHENDERSON ATCONTROL OF MARINE ORGANISMS IN A SALT*
648YEAGER WLCASTELLI VJ ANTIFOULING APPLICATIONS OF VARIOUS T*
649ADERSEN DM ROGANOTIN PRESERVATIVES FOR WOOD STRU*
650EKAMA HCVAN LONDENAMDE WOLF P RESULTS OF AN INQUIRY INTO THE CONDIT*
651ARNOLD MHCLARKE HJ PROJECT B:ULTRASONIC ANTIFOULING SHIP*
652ANON PROGRESS REPORT ON THE TECHNITIUM PRO*
653CHET I MITCHELL R THE RELATIONSHIP BETWEEN CHEMICAL STR*
654CLARKE GL POISONING AND RECOVERY IN BARNACLES A*
655CORNER EDSPARROW THE MODES OF ACTION OF TOXIC AGENTS-I*
656ANON THE 'TOXION' SYSTEM-A NEW ANTIFOULING *
657ANON REDUCING THE BARNACLE BILL
658ANON SHIPS' HULL PROTECTED-ULTRASONIC VIBR*
659FISK NR A VIEW OF ANTIFOULING
660FLEMING H EFFECT OF HIGH FREQUENCY FIELDS ON MI*

BLOCK OF 30
CARD #2

AD-8032 858 78
AD-918 043 73
AD-8006 675 74
AD-907 612 72
AD-902 136 72
AD-911 382 72
US PATENT# 4,052,354 77
AD-922 986 74
WATER SEWAGE WORKS,V96,N8,P279-83,A*79
NBS DIMENSIONS,V04,N2,P12-7,1980 80
APP MICROBIOL,V30,N2,P298-308,AUG 1975
COMP BIOCHEM PHYSIOL,V30,N6,P1037-4*69
NATURE,V196,N485,P539-41,NOV 10,196262
NAVSEA JOUR, 62-6,JULY 1976 76
TRANS ASME,V72,P127-31,FEB 1950 50
TRANS ASME,V22,P117-26,FEB 1950 50
J INST PET,V45,N426,P155-67,JUN 195959
ORGANOMETALLIC POLYMERS,ED CE CARRA*77
DTNSRDC REPORT # SME-78/41,JUN 1979 79
TNO REPORT 47C, 1962
TECHNICAL MINUTE #93 52
UNIV OF VIRGINIA ALUM PATENTS FOUND*79
CAN.J MICROBIOL,V22,P1206-08,N8,197676
BIOLOGICAL BULL,V92,P73-91,1947 47
J MAR BIOL ASS UK,V35,P531-48,1956 56
CORR PREV & CONT,P49-54,MARCH 1960 60
CHEM WEEK,V72,N9,87-91,FEB 28,1953 53
ENGINEERING,V180,P416,SEPT 23,1955 55
PAINT TECH,V24,N270,P15-18,MAY 1960 60
ELEC ENG,P18-21,JAN 1944 44



BLOCK OF 30
CARD #3

(blank card)
(blank card)
(blank card)
(blank card)
(blank card)

SECTION
DECK

3112
4110
4111
(blank card)
(blank card)
(blank card)

SAMPLE INPUT

OUTPUT FOR PRECEDING INPUT

1. POTENTIAL DUPLICATES (NOT INCLUDED)
2. ALPHABETIC SECTIONS REQUESTED (INCLUDED)
3. FULL ALPHABETIC REFERENCE LIST (NOT INCLUDED OVER 500 LINES OF OUTPUT)
4. FULL CHRONOLOGICAL REFERENCE LIST (NOT INCLUDED)
5. FULL NUMERICAL REFERENCE LIST (NOT INCLUDED)

ELASTOMERS SECTION REFERENCES

ACCESS NUMBER	FIRST AUTHOR	SECOND AUTHOR	THIRD AUTHOR	TITLE	PUBLISHER	YEAR
356	BINGHAM ,MH;	MUNN ,PW;	, ;	IMPRESSED CURRENT CATHODIC PROTECTION*	ANTI-CORROSIONV25 N12 P8-12	78
64	CASTELLI ,VJ;	MONTEMARA*,JA;	FISCHER ,EC;	ORGANOMETALLIC POLYMERS-ANTIFOULING M*	MARINE TECH SOC J VOL9 N7 P16	75
272	CASTELLI ,VJ;	, ;	, ;	CORROSION AND BIOFOULING ON THE NON-H*	FIRST ANN OTEC BIOFOUL AND CORR SYM*	77
289	CASTELLI ,VJ;	, ;	, ;	CORROSION AND BIOFOULING ON THE HEAT *	DTNSRDC REPORT #79/054,MAY 1979,82P	79
355	DEAR ,H ;	, ;	, ;	THE DESIGN AND APPLICATION OF ANTIFOU*	ADV ORGANIC COAT SCI TECH N79P152	79
152	DEFORST ,A ;	PETTIS ,RW;	PHILLIP ,AT;	ELASTOMERIC ANTIFOULING COATINGS *	AUST.MATERIALS RESEARCH LAB(TN378)	75
638	DE FORST ,A ;	PETTIS ,RW;	PHILLIP ,AT;	UNDERWATER MARINE COATINGS-A DETAILED*	AD-922 986	74
157	DICK ,RJ;	MERRILL ,BJ; *	, ;	EVALUATION OF PROTECTIVE COATINGS FOR*	DOT/USCG (REPT.CG-D-24-77) MAY 1977	77
464	EDELSTEIN ,HP;	ELLER ,SA;	GRUNTHER ,RG;	FOULING RESISTANT ELASTOMERIC MATERIA*	NAVAL ENG J V82 N1 2/70 P115-21	70
151	FISCHER ,EC;	BIRNBAUM ,LS; *	, ;	SURVEY REPORT: NAVY BIOLOGICAL FOULIN*	NAVAL UNDERSEA CENTER(NUC TP456)MAR	70
154	FROMER ,RL;	, ;	, ;	APPLICATION OF NOFOUL RUBBER BY B.F. *	US NAVY UNDERWATER SOUND LAB MAY1969	69
35	HOHMAN ,AE;	, ;	, ;	ELASTOMERIC COATINGS TO PROTECT AGAIN*	PROC 4TH INT CONG ON MAR COR & FOUL*	76
61	KRONSTEIN ,M ;	, ;	, ;	ENVIRONMENTALPROTECTREQMTSANTIFOULCOAT	MOD.PAINT COATINGS,DEC 1980,P45 47	80
108	KRONSTEIN ,M ;	, ;	, ;	CONTROLLED RELEASE OF POLYMERIC ORGAN*	ACS,DIV POLYMER CHEM PREPRINTV21N1	80
348	KUMAR ,A ;	WITTMER ,D ;	, ;	COATINGS AND CATHODIC PROTECTION OF P*	MATERIALS PERFORM V18 N12 P9-19	79
216	MAJOR ,CJ;	CARDARELLI,NF;	, ;	BIOCIDAL RUBBER FOR WATER RECLAMAION *	AERO MED RES LAB REPORT TR69-17 JUNE	69
590	MITCHELL ,R ;	BENSON ,PH;	, ;	MICRO- AND MACROFOULING IN THE OTEC P*	ARGONNE NAT LAB REPORT #ANL/OTEC-BC*	80
183	MOCK ,JA;	, ;	, ;	MARINE COATINGS SET A NEW COURSE	MATERIALS ENGINEERING VOL90N04 19	79
636	OCHILTREE ,BC;	, ;	, ;	ANTIFOULING ELASTOMERIC COMPOSITIONS	AD-911 382	72
67	PETTIS ,RW;	PHILLIP ,AT; *	, ;	ANTIFOULING ACTIVITY OF PHYTOTOXIC CO*	DOD MAT RES.LABS.AUSTRALIA MRLR698	77
634	PHILLIP ,AT;	, ;	, ;	UNDERWATER MARINE COATINGS- PART I- M*	AD-907 612	72
37	SHERRARD ,JR;	DICK ,RJ;	NOWACKI ,LJ;	NEW MARINE COATINGS TECHNOLOGY APPLIE*	PROC 4TH INT CONG ON MAR COR & FOUL*	76
354	STEELE ,MD;	DRISCO ,RW;	, ;	FUNGAL-RESISTANT ORGANOTIN PAINTS	JOUR COAT TECHNOL V48 N616 P59-63	76
254	THUST ,U ;	, ;	, ;	ORGANOTIN COMPOUNDS IN THE D.D.R. - P*	TIN AND ITS USES,N122,P3-5,1979	79
426	WOODFORD ,JM;	, ;	, ;	UNDERWATER MARINE COATINGS-PART 2 *	AUST DOD/DEF STD LAB(RPT496) 3/72	72

SCRUBBING SECTION REFERENCES

ACCESS NUMBER	FIRST AUTHOR	SECOND AUTHOR	THIRD AUTHOR	TITLE	PUBLISHER	YEAR
168	ANON J , ;	, ;	, ;	SHIP UNDERWATER MAINTENANCE,EVALUATIO*	US NAVY/NAVSEC (REPT 6136-77-9) 2/77	77
442	ANON K , ;	, ;	, ;	NEW UNDERWATER PROCESS CUTS HULL CLEA*	UNDERSEA TECHNOLOGY 9/69 P55	69
463	BENSON ,PH;	BRINING ,DL;	PERRIN ,DW;	MARINE FOULING AND ITS PREVENTION	MAR TECHNOL V10 N1 1/73 P30-37	73
273	BRASWELL ,JA;	LOTT ,DF;	HEDLICKA ,SM;	PRELIMINARY EVALUATION OF FLOW DRIVEN*	BIOFOULING,CORROSION AND MATERIALS *	79
272	CASTELLI ,VJ;	, ;	, ;	CORROSION AND BIOFOULING ON THE NON-H*	FIRST ANN OTEC BIOFOUL AND CORR SYM*	77
277	CASTELLI ,VJ;	FRITSCH ,AB;	ADAMSON ,WL;	AN EVALUATION OF SOME MECHANICAL CLEA*	PROC 5TH OTEC CONF,MIAMI BEACH,FLA *	78
289	CASTELLI ,VJ;	, ;	, ;	CORROSION AND BIOFOULING ON THE HEAT *	DTNSRDC REPORT #79/054,MAY 1979,92P	79
319	COLOGER ,CP;	BOHLANDER ,GS;	PREISER ,HS;	REVIEW OF UNDERWATER CLEANING METHODS*	J COAT TECHNOL V49 N682 PP51-55	77
424	FETKOVICH ,JG;	GRANNEMANN,GN;	* , ;	A STUDY OF FOULING&CORROSION PROBLEMS*	USA-DOE (RPT CO0-4041-9) 9/77	77
478	FRASER ,I ;	, ;	, ;	UNDERWATER HULL CLEANING THE SAVING	SHIPBLDG&SHIPG REC V113 N23 6/69	69
583	FREEMAN ,JH;	, ;	, ;	MARINE FOULING OF FIXED OFFSHORE INST*	CORR PREVENT CONTROL,V25,N6,P7-14,19	78
271	FRITSCH ,A ;	ADAMSON ,W ;	CASTELLI ,VJ;	AN EVALUATION OF MECHANICAL CLEANING *	FIRST ANN OTEC BIOFOUL AND CORR SYM*	77
86	GOODMAN ,EM;	GREENEBAUM,B ;	MARRON ,MT;	EFFECTS OF EXTREMELY LOW FREQUENCY EM*	RADIATION RESEARCH V66 P531-540 19	76
285	HAGEL ,D ;	CONN ,AF;	RICE ,MS;	METHODS FOR CLEANING OTEC HEAT EXCHAN*	PROC OTEC BIOFOULING AND CORR SYM, *	77
276	KERN ,WI;	, ;	, ;	INCREASING HEAT EXCHANGER EFFICIENCY *	PROC 4TH ANN CONF ON OTEC,NEW ORLEA*	77
587	KUESTER ,CK;	LYNCH ,CE;	, ;	AMERTAP AT ENGLISH STATION	WINTER ANN MEET ENER SYS EXP ASME, *	66
278	LEVENTHAL ,EL;	, ;	, ;	A BIOFOULING CONTROL SYSTEM FOR AN OT*	PROC 5TH OTEC CONF,MIAMI BEACH,FLA *	78
631	LORENZ ,J ;	, ;	, ;	ANTIFOULING MEASURES ON SHIPS-A GENER*	AD-B032 858	78
460	MALONE ,JA;	ALLMAN ,M ;	, ;	HULL PERFORMANCE ASSESSMENT MODEL VOL*	NTIS PB80-145816 (DOC-MARAD930-80015	80
112	MILNE ,A ;	, ;	, ;	HULL SURFACE MAINTENANCE-SMOOTHING TH*	CANADIAN SHIPP&MAR ENG VOL49 N04P18	78
590	MITCHELL ,R ;	BENSON ,PH;	, ;	MICRO- AND MACROFOULING IN THE OTEC P*	ARGONNE NAT LAB REPORT #ANL/OTEC-BC*	80
274	NUBEL ,ED;	, ;	, ;	AUTOMATIC TUBE CLEANING SYSTEM-BRUSH *	PROC 4TH ANN CONF ON OTEC,NEW ORLEA*	77
113	PASCOE ,DW;	, ;	, ;	CARWASH APPROACH HAS MERIT	CANADIAN SHIPP&MAR ENG VOL49 N04P18	78
14	PREISER ,HS;	BOHLANDER ,GS;	COLOGER ,CP;	FOULING CONTROL MEANS FUEL SAVINGS FO*	SNAME 'STAR' SYM,25MAY1977 SANFRANC*	77
51	PREISER ,HS;	COLOGER ,CP;	BOHLANDER ,GS;	UNDERWATER HULL CLEANING FOR FUEL CON*	RNZN COR & FOUL COM, PROC 5TH INTER*	77
245	SATO ,S ;	NAGATA ,K ;	OGISO ,A ;	EFFECT OF SPONGE BALL CLEANING ON COR*	SUMITOMO LIGHT METAL TECHNICAL REPO*	72
284	SCHLESING&,HA;	, ;	, ;	ECONOMICS OF ALTERNATIVES FOR OTEC BI*	PROC OTEC BIOFOULING AND CORR SYM, *	77
425	SPOONER ,CM;	, ;	, ;	ASSESSMENT OF CORROSION PRODUCTS FROM*	USA-EPA (RPT EPA600/7-80-026) 1/80	80
569	TROTMAN ,DW;	, ;	, ;	UNDERWATER CLEANING	PROC 1ST INTER SHIP PAINT CORR CONF*	74
570	TROTMAN ,DW;	JACKSON , ;	, ;	UNDERWATER CLEANING	PROC 1ST INTER SHIP PAINT CORR CONF*	74
75	VAN LONDEN,AM;	JOHNSEN ,S ;	GOVERS ,GJ;	THE CASE OF LONG-LIFE ANTIFOULINGS	J PAINT TECHNOL V47 N600 P62-68	75
87	WAYLAND ,JR;	, ;	, ;	THEORETICAL APPROACH TO THE EFFECTS O*	RADIATION RESEARCH V74 P207-216 19	78

EXTERIOR SECTION REFERENCES

ACCESS NUMBER	FIRST AUTHOR	SECOND AUTHOR	THIRD AUTHOR	TITLE	PUBLISHER	YEAR
272	CASTELLI ,VJ;	, ;	, ;	CORROSION AND BIOFOULING ON THE NON-H%	FIRST ANN OTEC BIOFOUL AND CORR SYM%	77
289	CASTELLI ,VJ;	, ;	, ;	CORROSION AND BIOFOULING ON THE HEAT %	DTNSRDC REPORT #79/054,MAY 1979,82P	79
565	DRAKE ,RC;	, ;	, ;	INCREASING HEAT EXCHANGER EFFICIENCY %	BIOFOULING CONTROL PROCEDURES,POLLU%	77
583	FREEMAN ,JH;	, ;	, ;	MARINE FOULING OF FIXED OFFSHORE INST%	CORR PREVENT CONTROL,V25,N6,P7-14,19	78
590	MITCHELL ,R ; BENSON ,PH;		, ;	MICRO- AND MACROFOULING IN THE OTEC P%	ARGONNE NAT LAB REPORT #ANL/OTEC-BC%	80
14	PREISER ,HS; BOHLANDER ,GS;	COLOGER ,CP;	BOHLANDER ,GS;	UNDERWATER HULL CLEANING FOR FUEL CON%	SNAME 'STAR' SYM,25MAY1977 SANFRANC%	77
51	PREISER ,HS; COLOGER ,CP;	BOHLANDER ,GS;	, ;	UNDERWATER CLEANING	RNZN COR & FOUL CON, PROC 5TH INTER%	77
569	TROTMAN ,DW;	, ;	, ;	UNDERWATER CLEANING	PROC 1ST INTER SHIP PAINT CORR CONF%	74
570	TROTMAN ,DW; JACKSON , ;		, ;	UNDERWATER CLEANING	PROC 1ST INTER SHIP PAINT CORR CONF%	74
75	VAN LONDEN,AM; JOHNSEN ,S ;	GOVERS ,GJ;		THE CASE OF LONG-LIFE ANTIFOULINGS	J PAINT TECHNOL V47 N600 P62-68	75

APPENDIX C

AUXILIARY PROGRAMS

Several auxiliary programs proved helpful in editing the references file (PREFS). Most editing was performed by the use of the on-line editor, NETED, v. 1.4, maintained on the DTNSRDC computer system. In the event an entry was duplicated, the duplicate was deleted through the use of the editor. Since there were several individuals submitting data to this bibliography, it was desired that all sections marked by each author be combined into the calculated index on the remaining entry. A method was required to do this calculation.

Program NEWIND will combine two indices and give a new index, to be inserted into the reference file, which will cover all sections marked by both authors.

Program INDICES will back calculate and give a listing of all sections in a single index or a group of indices. This was useful in comparing what sections were marked for the same article by two different authors. For 41 duplicates with a total of 278 sections marked, 41% of the sections were marked by both authors. This shows a definite advantage to examining references on more than one occasion.

Additionally, two programs were written to help in formatting the bibliographic entries. Program WRITE will write out the permanent reference file (maintained in alphabetic order) in an expanded format to allow for completion of the final bibliography. It also numbers the entries so that a record of the size of the bibliography is maintained. Program LIST will numerically sort the references and give the list out in appropriate format for the citations to be placed in the text of the article or book (i.e., Jones, 1966).

Program SECCOR is used as an interactive editor to correct the sections a reference is found in. By connecting the files INPUT, OUTPUT, and TTY to a terminal, corrections are easily made with interactive prompts. The program will ask for the access number and then give the author and section index for that access number. The program will then ask for a section number and if the section is to be inserted or deleted. It is not necessary to know the status of the section in the original index, the program will check that status. It will continue to ask for sections until the number "Ø" is typed in, at that point it will ask for a new

access number. A response of "Ø" to the access number will stop the program. A new file of the references, with the corrected section index numbers, will be created under file name NEWREF. This file must then be cataloged under the permanent reference file name PREFS. This program requires that the permanent reference file be maintained under file name PREFS.

```
      PROGRAM NEWIND (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C*************************************************************************
C
C THIS PROGRAM WILL COMBINE TWO INDICES AND GIVE A NEW INDEX.
C IT IS ESSENTIALLY DOING A LOGICAL AND ON TWO 40 BIT WORDS.
C
C*************************************************************************
C
    1 READ (5,100) IND1
      IF (IND1.EQ.0) STOP
      READ (5,100) IND2
  100 FORMAT (I13)
      P=IND1
      P1=IND2
      IND=0
C
C*************************************************************************
C
C CHECK IF THE BIT IS SET ON EITHER WORD AND SET THE NEW BIT IF THIS
C CONDITION IS MET.
C
C*************************************************************************
C
      DO 15 K=1,40
      L=40-K
      M=P-2**L
      M1=P1-2**L
      IF (M.GE.0.OR.M1.GE.0) IND=IND+2**L
      IF (M.GE.0) P=M
      IF (M1.GE.0) P1=M1
   15 CONTINUE
C
C*************************************************************************
C
C PRINT OUT THE OLD INDICES AND THE NEW COMBINED INDEX.
C
C*************************************************************************
C
      WRITE (6,110) IND1,IND2,IND
  110 FORMAT ('0','THE OLD INDICES WERE ', I13,' AND ',I13,6X,'THE NEW '
     A,'COMBINED INDEX IS ',I13)
      GO TO 1
      END
```

INPUT

```
  26791580349
    469108736
  16891348565
    468703594
0000
0000
```

OUTPUT

```
THE OLD INDICES WERE   26791580349 AND     469108736     THE NEW COMBINED INDEX IS   26842962621
THE OLD INDICES WERE   16891348565 AND     468703594     THE NEW COMBINED INDEX IS   17178812287
```

NEWIND INPUT AND OUTPUT

```
      PROGRAM INDICES (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C*********************************************************************
C
C     THIS PROGRAM GIVES THE SECTION NUMBERS RELATED TO ANY INPUT
C     INDEX.  THE APPLICABLE SECTION NUMBERS ARE DESIGNATED BY A '1'
C     THE INDEX NUMBERS WHICH ARE DESIRED TO BE BACK CALCULATED SHOULD
C     BE INPUT ON CONSECUTIVE CARDS FOLLOWED BY A BLANK CARD
C     THE PROGRAM ACCEPTS THE DATA IN I13 FORMAT AND LOOKS FOR A ZERO
C     AS A FLAG FOLLOWING THE LAST DATA ENTRY.
C
C*********************************************************************
C
C
      COMMON INDEX(40)
      CALL WRT
      CONTINUE
    1 READ (5,100)IBINX
  100 FORMAT (I13)
      IF (IBINX.EQ.0) STOP
      P=IBINX
C
C*********************************************************************
C
C CHECK WHICH BITS ARE SET AND INDICATE BY PUTTING ONES IN AN
C ARRAY.
C
C*********************************************************************
C
      DO 5 I=1,40
      INDEX (I)=0
    5 CONTINUE
      DO 15 K=1,40
      L=40-K
      M=P-2**L
      N=41-K
      IF (M.LT.0) GO TO 25
      INDEX (N)=1
      P=M
   25 CONTINUE
   15 CONTINUE
      WRITE (6,140) IBINX,(INDEX(I),I=1,40)
  140 FORMAT ('0',I13,1X,40I2)
      GO TO 1
      END
```

```
      SUBROUTINE WRT
C
C*********************************************************************
C
C THIS SUBROUTINE WRITES OUT THE HEADINGS WHICH ARE THE SECTION
C NUMBERS TO WHICH THE SET BITS CORRESPOND.
C
C*********************************************************************
C
      COMMON INDEX (40)
      WRITE (6,100)
  100 FORMAT('1',15X,'1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3',
     A' 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5')
      WRITE (6,110)
  110 FORMAT (16X,'0 1 0 1 1 2 2 2 2 0 1 1 1 1 1 1 2 2 0 1 1 1',
     A' 1 1 1 1 1 1 2 3 4 5 6 7 8 9 0 1 2')
      WRITE (6,120)
  120 FORMAT (16X,'0 0 0 0 1 0 1 2 3 0 0 1 1 1 1 1 1 0 1 0 0 1 1 1',
     A' 2 3 3 3 4 0 0 0 0 0 0 0 0 0 0 0')
      WRITE (6,130)
  130 FORMAT (1X,'INDEX',10X,'0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5',
     A' 0 0 0 0 0 1 2 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0 0')
      RETURN
      END
```

INPUT

```
26791580349
  469108736
16891348565
  468703594
0000
```

OUTPUT

INDEX	1 0 0 0	1 1 0 0	2 0 0 0	2 1 0 0	2 1 1 0	2 2 0 1	2 2 1 2	2 2 2 3	3 0 0 0	3 1 0 1	3 1 1 1	3 1 1 2	3 1 1 3	3 1 1 4	3 1 1 5	3 2 0 0	3 2 1 0	4 0 0 0	4 1 0 0	4 1 1 0	4 1 1 1	4 1 1 2	4 1 2 0	4 1 3 0	4 1 3 1	4 1 3 2	4 1 4 0	4 2 3 0	4 3 4 0	4 4 5 0	4 5 6 0	4 6 7 0	4 7 8 0	4 8 9 0	5 0 1 0	5 1 2 0
26791580349	1	0	1	1	1	1	0	1	0	1	0	0	1	0	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0
469108736	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
16891348565	1	0	1	0	1	0	1	0	0	1	1	0	0	0	0	1	1	0	1	1	0	0	1	1	0	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0
468703594	0	1	0	1	0	1	1	0	1	0	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0

INDICES INPUT AND OUTPUT

```
      PROGRAM WRITE (PREFS,OUTPUT,TAPE1=PREFS,TAPE2=OUTPUT)
C
C*********************************************************************
C
C     THIS PROGRAM IS DESIGNED TO REWRITE AND RENUMBER A BIBLIOGRAPHIC
C     FILE INTO A FORM SUITABLE FOR EDITING PRIOR TO TYPING
C
C*********************************************************************
C
      COMMON LIST (1000,17)
      DO 5 I=1,1000
      READ (1,100) (LIST(I,K),K=2,17)
  100 FORMAT (I6,A10,A2,A10,A2,A10,A2,3A10,A8,3A10,A6,I2)
      IF (LIST(I,2).EQ.0) GO TO 10
      LIST (I,1)=I
      WRITE (2,110) (LIST(I,K),K=1,17)
  110 FORMAT ('*',I6,2X,I6,2X,A10,',',1X,A2,',',2X,A10,',',1X,A2,',',2X,
     AA10,',',1X,A2,',',15X,3A10,A8/,'*',3A10,A6,70X,'19',I2,/)
    5 CONTINUE
   10 STOP
      END
```

```
      PROGRAM LIST (PREFS,TTY,OUTPUT,TAPE5=PREFS,TAPE6=TTY,TAPE7=OUTPUT)
C
C*********************************************************************
C
C  THIS PROGRAM IS DESIGNED TO INTERPRET THE REFERENCE FILE AND
C  PLACE THE REFERENCES IN A FORM APPROPRIATE FOR THE
C  CITATIONS IN THE TEXT OF THE PAPER.
C
C*********************************************************************
C
      COMMON IWKLST (850,5)
      N=0
      DO 5 I=1,850
      READ (5,100)(IWKLST(I,J),J=1,5)
  100 FORMAT(I6,A10,2X,A10,2X,A10,76X,I2)
      IF (IWKLST(I,1).EQ.0) GO TO 10
      N=N+1
    5 CONTINUE
   10 CONTINUE
      M1=N
   35 I2=0
      M1=M1-1
      DO 15 I=1,M1
      I1=IWKLST(I,1)
      J=I+1
      I3=IWKLST(J,1)
      IF (I3-I1) 20,25,25
   20 DO 30 K=1,5
      M=IWKLST(I,K)
      IWKLST(I,K)=IWKLST(J,K)
      IWKLST(J,K)=M
   30 CONTINUE
      I2=I2+1
   25 CONTINUE
   15 CONTINUE
      IF(I2.GT.0)GO TO 35
      DO 40 I=1,N
      IF (IWKLST(I,4).GE.0) GO TO 50
      IF (IWKLST(I,3).GE.0) GO TO 60
      WRITE (6,110) IWKLST(I,1),IWKLST(I,2),IWKLST(I,5)
  110 FORMAT (5X,I6,10X,A10,',',1X,'19',I2)
      GO TO 45
   50 WRITE (6,120) IWKLST(I,1),IWKLST(I,2),IWKLST(I,5)
  120 FORMAT (5X,I6,10X,A10,' ET AL,',1X,'19',I2)
      GO TO 45
   60 WRITE (6,130)IWKLST(I,1),IWKLST(I,2),IWKLST(I,3),IWKLST(I,5)
  130 FORMAT(5X,I6,10X,A10,' AND ',A10,',',1X,'19',I2)
   45 CONTINUE
   40 CONTINUE
      STOP
      END
```

```
      PROGRAM SECCOR (OUTPUT,TTY,PREFS,INPUT,NEWREF,TAPE1=OUTPUT,
     ATAPE2=TTY,TAPE3=PREFS,TAPE4=INPUT,TAPE5=NEWREF)
C
C*********************************************************************
C
C    THIS PROGRAM IS DESIGNED AS AN INTERACTIVE METHOD TO CORRECT
C    THE SECTIONS A REFERENCE IS FOUND IN.  THE FILES INPUT,
C    OUTPUT AND TTY MUST BE CONNECTED TO YOUR TERMINAL AND THE FILE
C    PREFS, PERMANENT REFERENCES, MUST BE AVAILABLE TO THE
C    PROGRAM AS THE WORKING DATA FILE.
C
C*********************************************************************
C
      COMMON LIST(1000,14),INDEX(40),IBIT(40)
      DO 10 I=1,1000
      READ (3,1) (LIST(I,J),J=1,14)
    1 FORMAT (I6,A10,10A10,A2,I13)
      IF (LIST(I,1).EQ.0) GO TO 20
   10 CONTINUE
   20 CONTINUE
      DO 12 I6=1,40
      I7=I6-1
      IBIT(I6)=2**I7
   12 CONTINUE
C
C*********************************************************************
C
C    IDENTIFY REFERENCE YOU WISH TO AMEND, INSERT 00 IF YOU WISH
C    TO END PROGRAM.  INPUT MUST BE IN I6 FORMAT, IE 111 IS INPUT
C    AS 000111.
C
C*********************************************************************
C
    2 PRINT (2,5)
    5 FORMAT ('0','TYPE IN ACCESS NO. IN I6 FORMAT.  ')
      READ (4,15) IAC
   15 FORMAT (I6)
      IF (IAC.EQ.0) GO TO 4
      IND=0
      DO 30 I=1,1000
      IF (IAC.EQ.LIST(I,1)) WRITE (2,25)(LIST(I,J),J=1,2),LIST(I,14)
   25 FORMAT ('0','*',I6,2X,'FIRST AUTHOR ',A10,2X,'INDEX  ',I13)
      IF (IAC.EQ.LIST(I,1)) NUM=I
      IF (IAC.EQ.LIST(I,1)) IND=LIST(I,14)
      IF (LIST(I,1).EQ.0)GO TO 40
   30 CONTINUE
   40 CONTINUE
```

```
C
C*********************************************************************
C
C   WAS REFERENCE FOUND?  IF NOT, ASK FOR ACCESS NUMBER AGAIN.
C
C*********************************************************************
C
      IF (IND.EQ.0) GO TO 2
      IND1=IND
C
C*********************************************************************
C
C   TRANSLATE INDEX NUMBER OF REFERENCE IN SECTIONS PREVIOUSLY SET.
C
C*********************************************************************
C
      DO 55 I=1,40
      N=41-I
      INDEX(N)=0
      M=IND1-IBIT(N)
      IF (M.LT.0) GO TO 65
      INDEX(N)=1
      IND1=M
   65 CONTINUE
   55 CONTINUE
C
C*********************************************************************
C
C   IDENTIFY SECTION YOU WANT CHANGED.  INSERT 00 IF YOU ARE
C   FINISHED WITH THIS ACCESS NUMBER.
C
C*********************************************************************
C
    3 PRINT (2,35)
   35 FORMAT ('0','TYPE IN SECTION NUMBER   ')
    8 READ (4,45) ISECT
      IF (ISECT.EQ.0) GO TO 2
   45 FORMAT (I4)
      CALL SECTIN (ISECT,I1)
      IF (I1.EQ.0) PRINT (2,100)
  100 FORMAT ('0','ERROR IN SECTION NUMBER, RE-ENTER  ')
      IF (I1.EQ.0) GO TO 8
      I5=I1-1
```

```
C
C*************************************************************************
C
C    DO YOU WANT SECTION ADDED OR REMOVED?
C
C*************************************************************************
C
      PRINT (2,75)
   75 FORMAT ('0','TYPE IN 1 FOR INSERTION, 2 FOR DELETION  ')
      READ (4,85) IFLAG
   85 FORMAT (I1)
      IF (IFLAG.EQ.2) GO TO 6
C
C*************************************************************************
C
C    CHECK IF SECTION IS ALREADY PRESENT.  IF NOT, ADD
C    APPROPRIATE FACTOR TO THE SECTION INDEX.
C
C*************************************************************************
C
      IF (INDEX(I1).EQ.0) IND=IND+IBIT(I1)
      LIST(NUM,14)=IND
      GO TO 3
C
C*************************************************************************
C
C    CHECK IF SECTION IS NOT PRESENT.  IF PRESENT, SUBTRACT
C    APPROPRIATE FACTOR FROM THE SECTION INDEX.
C
C*************************************************************************
C
    6 IF (INDEX(I1).EQ.1) IND=IND-IBIT(I1)
      LIST(NUM,14)=IND
      GO TO 3
C
C*************************************************************************
C
C    WRITE OUT A NEW CORRECTED REFERENCE FILE TO FILE NEWREF.
C
C*************************************************************************
C
    4 DO 50 I=1,1000
      WRITE (5,95)(LIST(I,J),J=1,14)
   95 FORMAT (I6,A10,10A10,A2,I13)
      IF (LIST(I,1).EQ.0) STOP
   50 CONTINUE
      END
```

```
      SUBROUTINE SECTIN (ISECT,I1)
      COMMON LIST (1000,14),INDEX (40),IBIT(40)
C
C*********************************************************************
C
C    THIS SUBROUTINE IDENTIFIES THE SECTION YOU ARE REFERRING TO,
C    FOR COMPUTER USAGE.
C
C*********************************************************************
C
      I1=0
      IF (ISECT.EQ.1000) I1=1
      IF (ISECT.EQ.1100) I1=2
      IF (ISECT.EQ.2000) I1=3
      IF (ISECT.EQ.2100) I1=4
      IF (ISECT.EQ.2110) I1=5
      IF (ISECT.EQ.2200) I1=6
      IF (ISECT.EQ.2210) I1=7
      IF (ISECT.EQ.2220) I1=8
      IF (ISECT.EQ.2230) I1=9
      IF (ISECT.EQ.3000) I1=10
      IF (ISECT.EQ.3100) I1=11
      IF (ISECT.EQ.3110) I1=12
      IF (ISECT.EQ.3111) I1=13
      IF (ISECT.EQ.3112) I1=14
      IF (ISECT.EQ.3113) I1=15
      IF (ISECT.EQ.3114) I1=16
      IF (ISECT.EQ.3115) I1=17
      IF (ISECT.EQ.3200) I1=18
      IF (ISECT.EQ.3210) I1=19
      IF (ISECT.EQ.4000) I1=20
      IF (ISECT.EQ.4100) I1=21
      IF (ISECT.EQ.4110) I1=22
      IF (ISECT.EQ.4111) I1=23
      IF (ISECT.EQ.4112) I1=24
      IF (ISECT.EQ.4120) I1=25
      IF (ISECT.EQ.4130) I1=26
      IF (ISECT.EQ.4131) I1=27
      IF (ISECT.EQ.4132) I1=28
      IF (ISECT.EQ.4140) I1=29
      IF (ISECT.EQ.4200) I1=30
      IF (ISECT.EQ.4300) I1=31
      IF (ISECT.EQ.4400) I1=32
      IF (ISECT.EQ.4500) I1=33
      IF (ISECT.EQ.4600) I1=34
      IF (ISECT.EQ.4700) I1=35
      IF (ISECT.EQ.4800) I1=36
      IF (ISECT.EQ.4900) I1=37
      IF (ISECT.EQ.5000) I1=38
      IF (ISECT.EQ.5100) I1=39
      IF (ISECT.EQ.5200) I1=40
      RETURN
      END
```

INITIAL DISTRIBUTION

Copies		Copies	Code	Name
2	Defense Documentation Center	2	28	J. R. Belt
		2	284	A. Rufolo
		15	2844	E. C. Fischer
		1	522.1	Unclass Library
		2	5231	Office Services

DTNSRDC ISSUES THREE TYPES OF REPORTS

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.

2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.

3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.


DA
FILM